Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

“САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ”

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

РАСПОЗНАВАНИЕ ИМЕНОВАННЫХ СУЩНОСТЕЙ В РУССКОМ ЯЗЫКЕ С ИСПОЛЬЗОВАНИЕМ ГЛУБОКИХ НЕЙРОННЫХ СЕТЕЙ

Автор *Коноплич Георгий Викторович* _____

Направление подготовки *01.04.02 Прикладная математика и информатика*

Квалификация Магистр

Руководитель *Фильченков А. А., к.ф.-м.н., доц. каф. КТ* _____

К защите допустить

Зав. кафедрой КТ *Васильев В.Н., проф., д.т.н.* _____

“___” _____ 2018 г.

Санкт-Петербург, 2018 г.

Студент *Коноплич Г.В.* Группа *М4238* Кафедра *КТ* Факультет *ИТиП*

Направленность (профиль), специализация
Технологии проектирования и разработки программного обеспечения

Консультант(ы):

а) *Путин Е.О., программист кафедры ИФ, аспирант КТ* _____

ВКР принята «____» _____ 2018 г.

Оригинальность ВКР _____%

ВКР выполнена с оценкой _____

Дата защиты «____» июня 2018 г.

Секретарь ГЭК *Павлова О.Н.* _____

Листов хранения _____

Демонстрационных материалов/Чертежей хранения *отсутствуют*

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

В настоящее время происходит экспоненциальный рост объёмов информации в неструктурированной форме. Всемирная сеть состоит в основном из неструктурированных документов, из которых тяжело получить полезную информацию. Человеческих ресурсов для анализа и обработка такого большого количества информации катастрофически недостаточно. Учитывая всё это, наличие алгоритмов и методов, способных производить анализ информации, структурировать и представлять её в понятном человеку виде, становится критической задачей. Знания, содержащиеся неструктурированных документах, могут быть доступны для машинной обработки и приведения их в структурированную форму.

Современное развитие искусственного интеллекта позволяет успешно решать ряд задач в области обработки естественного языка без участия человека. Обработка естественного языка (natural language processing, NLP) – это важная область исследований в компьютерных науках, занимающиеся анализом документов на основе множества теорий и технологий. Важной частью обработки естественного языка является извлечение информации. Извлечение информации из текста заключается в извлечении объектов, связи между этими объектами и в конечном итоге в извлечении необходимых фактов. Это процесс можно описать как понимание текста без участия человека.

Сейчас активно развиваются диалоговые системы, в которых понимание текста является одной из главных задач. Искусственный интеллект в области обработки естественного языка может помогать операторам технической поддержки находить быстрее правильный ответ или общаться с людьми на начальной стадии диалога. Также на новостных порталах важно держать все статьи в структурированном виде, искусственный интеллект может сам классифицировать документ на нужную тему либо предложить ряд подходящих тем на выбор человеку. В социальных сетях и прочих источниках, содержащих развлекательный контент, пользователю постоянно надо рекомендовать что-то новое, а иногда и то, о чём он сам не догадывается. Искусственный интеллект

может помочь человеку в изучении иностранных языков, в сортировке писем на почте, отвечать на некоторые письма. Голосовые команды сейчас внедряются повсюду, а это тоже лежит в области обработки естественного языка. В конечном итоге хочется прийти к тому, чтобы искусственный интеллект понимал естественный язык так же хорошо, как и человек. Мог применять знания мира при их анализе. На данный момент это актуальная и сложная задача.

Для создания всего вышеописанного почти всегда сперва необходимо решить задачу извлечения объектов из текста или, другими словами, распознать именованные сущности в тексте (named entity recognition, NER).

Целью данной работы является разработка алгоритмов на основе глубоких нейронных сетей для решения задачи распознавания именованных сущностей в русском языке и достичь наилучших результатов.

В первой главе будет описание области обработки естественного языка. Будут приведены типы решаемых задач и некоторые примеры. Трудности, с которыми сталкиваются алгоритмы при решении задач в русском языке. Будет описана история появления задачи распознавания именованных сущностей, самые известные соревнования и конференции на которых она решалась. Будут описаны несколько типов классических сущностей, приведены методы решения вышеописанной задачи с помощью существующих ручных правил, классических статистических алгоритмов, алгоритмов на основе нейронных сетей и гибридных систем (сочетание нескольких видов алгоритмов вместе).

Во второй главе будет приведено описание спроектированных и использованных алгоритмов и подходов. Архитектура рекуррентной нейронной сети с морфологическими и синтаксическими признаками на вход, выделенными открытыми инструментами для обработки естественного языка. Далее спроектированная двунаправленная рекуррентная нейронная сеть, в качестве признаков к которой используются векторные представления слов из различных дистрибутивных семантических моделей русского языка, признаки с выхода другой нейронной рекуррентной сети, а также морфологические и синтаксические признаки, выделенные в предыдущем подходе. Будет описана

двунаправленная языковая модель, на внутренних слоях которой кодируются глубокие знания естественного языка. Будет описан метод тонкой настройки нейронной сети языковой модели. Приведен метод получения глубоких контекстно-зависимых векторных представлений слов из двунаправленной языковой модели, которые решают недостатки предыдущих подходов.

В третьей главе будут описаны детали реализации спроектированных алгоритмов. Будет приведено описание выбранных наборов данных. Описаны детали экспериментов с существующими и разработанными алгоритмами, а также метрики качества, использующиеся для тестирования моделей. Будет продемонстрировано сравнение с другими подходами.

ГЛАВА 1. ОБРАБОТКА ЕСТЕСТВЕННОГО ЯЗЫКА

История области обработки естественного языка началась еще в 1950-х годах. Однако, вплоть до 1980-х годов большинство систем обработки естественного языка были реализованы на наборах сложных рукописных правил. С внедрением методов машинного обучения произошла революция в этой области. Одним из ранних методов использовались деревья решения [36], которые были альтернативой рукописным правилам. Также стали популярны статистические модели, такие как скрытая марковская модель [35]. Такие модели были более надёжны на неизвестных данных и на данных, которые содержали ошибки (как это часто встречается в реальности). Сейчас очень популярны решения, основанные на обучении без учителя или с частичным привлечением учителя [37]. Такие алгоритмы могут извлекать знания из неаннотированных текстов, что очень важно, если учесть рост объёма информации. Глубокие нейронные сети стали неотъемлемой частью обработки естественного языка и позволяют достичь наилучших результатов.

Область обработки естественного языка решает множество задач:

- машинный перевод – это область вычислительной лингвистки, что исследует применение программного обеспечения для перевода текста с одно языка на другой. В простом приближении это простая замена слов одного языка на другой;
- вопрос-ответные системы – системы, которые могут отвечать на вопросы. Для поиска ответа, к примеру, используется Википедия. По входному вопросу одна нейронная сеть сужает список статей, в которых искать, а другая уже делает подробный поиск ответа;
- распознавание речи – это область вычислительной лингвистики, которая изучает применение технологий для распознавания разговорного языка и перевода его в текст;
- классификация текстов – определение тем документов. Важная задача в доменных областях;

- извлечение фактов – процесс анализа документов на основе множества теорий и технологий. Извлечение объектов, связи между этими объектами и в конечном итоге в извлечении необходимых фактов;
- диалоговые системы – это компьютерные системы, предназначенные для общения с человеком. Эти системы состоят из множества компонентов: распознавания речи, жестов, понимания языка (понимание языка уже сама по себе комплексная задача), выдача заготовленного или сгенерированного ответа, перевод в разговорный язык или на язык жестов;
- анализ тональности текста – определение эмоциональной окраски, настроения текста;
- суммаризация текстов – процесс сокращения текстов, извлечения только важной информации;
- topic modeling – определение списка тем из большой коллекции документов. Каждая тема представляется распределением слов в ней;
- разрешение кореферентности – определение связей между объектами и компонентами высказывания;
- разрешение лексической многозначности – определение смысла слова в зависимости от контекста;
- определение частей речи;
- синтаксический разбор;
- приведение слова в начальную форму;

Задачи можно разбить на несколько категорий: морфологические, синтаксические, семантические и прагматические. Первые три категории задач относятся к грамматике языка – науке, изучающей закономерности построения правильных и осмысленных предложений и текстов. Прагматика же изучает отношение знаков к субъектам, которые их воспроизводят и интерпретируют. Эффективные решения подобных задач разрабатывать сложнее всего. На

рисунке 1 представлена пирамида задач: чем выше уровень, тем больше сложность задач.

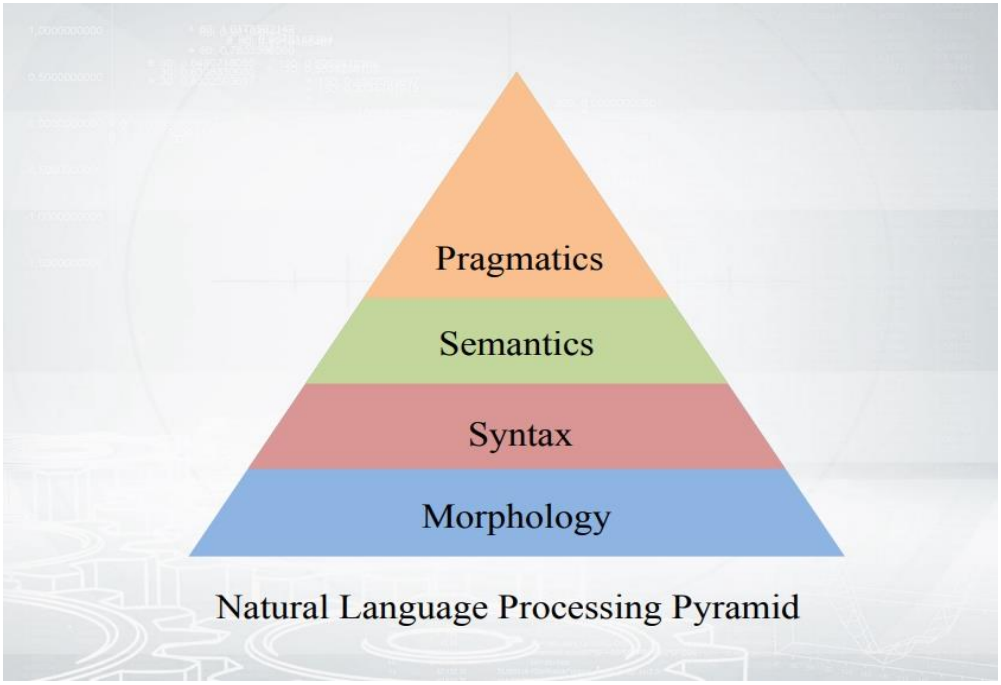

Рис. 1. Пирамида задач в обработке естественного языка

Сложности, возникающие при обработке естественного языка:

- **Анафоры** – определение к какому существительному относится местоимение. Например: «Антон поехал в свой дом на машине, потому что он устал. Антон поехал в свой дом на машине, потому что он находится далеко». В первом случае «он» — это «Антон», а во втором случае «он» — это «дом».
- **Свободный порядок слов** – «Дети играют на улице. Играют на улице дети».
- **Неологизмы –** новые слова в языке.
- **Полисемия –** множество значений у одного слова (омонимы, омографы).
- **И многое другое.**

.

1.1. История и суть задачи распознавания именованных сущностей

На шестой конференции Message Understanding (MUC-6) в 1996 году задачи фокусировались в области извлечения информации. В процессе постановки задач выявилась отдельная задача в извлечении объектов из документов. Для определения объекта ввели термин «именованная сущность», а задачу назвали распознавание именованных сущностей (named entity recognition, NER), так её в области обработки естественного языка и называют до сих пор. Сейчас необходимо было уметь распознавать в тексте различные сущности: имена, организации, числовые выражения, местоположения, даты, время и многое другое.

Не так давно на международной конференции по компьютерной лингвистике «Диалог 2016» проводилось соревнование по извлечению информации из текстов на русском языке [16]. Был подготовлен размеченный набор данных из новостных коллекций и решалась задача распознавания именованных сущностей. Необходимо было распознать следующие классические именованные сущности:

1. Персона (person, PER).
2. Местоположение (location, LOC).
3. Организация (organization, ORG).
4. Остальное (O) – тег для остальных слов в тексте, которые не являются обозначенными выше именованными сущностями.

1.2. Существующие методы решения

1.2.1. Ручные правила. Онтологии. Статистические методы. Классические методы машинного обучения

Для решения задачи распознавания именованных сущностей используют подходы, основанные на рукописных правилах. При таком подходе достаточно легко разобраться в причинах ошибки и исправить её, однако, процесс подбора исчерпывающего подбора правил очень ресурсоёмкий. Придумывать правила легче всего для классических именованных сущностей: имена (персоны), организации, местоположения. Однако, правила для всех ситуаций не придумаешь.

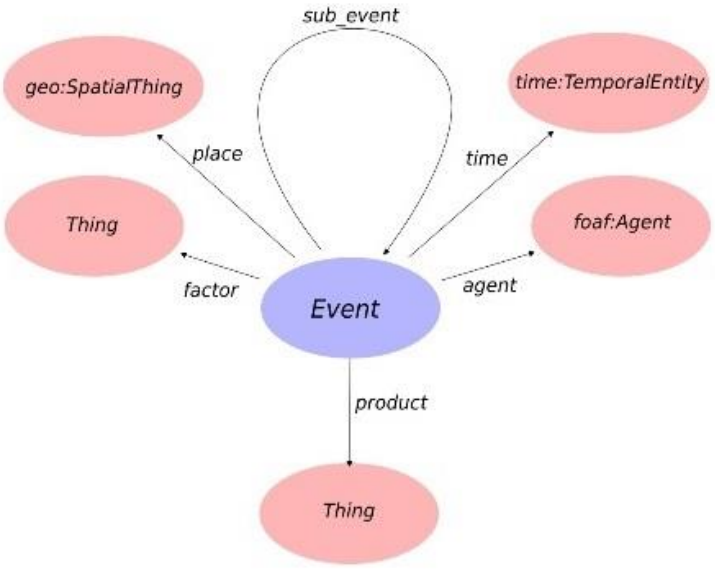

Рис. 2. Пример Онтологии

Также решать задачу NER можно с помощью онтологий (рис. 2). Онтология — это структура данных, содержащая понятия и объекты, правила и отношения между ними.

Берём онтологию Википедии и ищем в ней информацию об объектах. На рисунке 3 слева в предложении выделены именованные сущности, которые мы распознали с помощью онтологии Википедии.

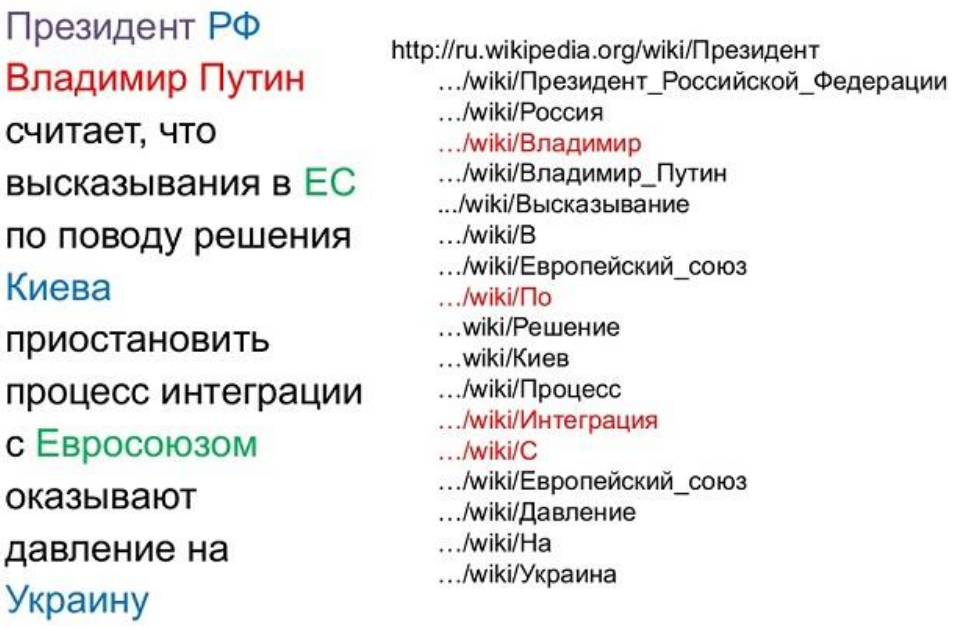

Рис. 3. Пример с онтологией Википедии

С применением онтологий достигаются хорошие точности в распознавании именованных сущностей, но такие системы совсем не имеют никакой обобщающей способности, их применение в другой доменной области или с новыми возникающими объектами не принесёт никаких результатов.

Также при решении задачи применяют алгоритмы машинного обучения. Признаки для таких алгоритмов нужно придумывать вручную. Классическим алгоритмом для решения NER могут быть деревья принятия решений – альтернатива рукописным правилам, алгоритм сам на основе признаков выучивает наилучшие правила. Популярным статистическими моделями являются скрытые марковские цепи или условные случайные поля. Сложности в таких подходах встречаются при извлечении хороших признаков. Такие системы часто страдают от переобучения.

1.2.2. Методы на основе нейронных сетей. Гибридные системы

В настоящее время очень популярны решения, основанные на нейронных сетях. Открытым стоит вопрос выбора признаков для нейронной сети: ручные правила, морфологические, синтаксические и семантические. Неотъемлемой

частью задач обработки естественного языка стало использование векторного представления слов, полученного из различных обученных моделей языка.

Как известно, нейронная сеть умеет хорошо автоматически извлекать репрезентативные признаки. Также рекуррентная нейронная сеть кодирует входную последовательность слов в контекстно-чувствительное представление. Однако, для качественного представления нейронной сети нужна большая обучающая выборка. Для решения задачи распознавания именованных сущностей современная архитектура нейронной сети состоит обычно из нескольких двунаправленных рекуррентных слоёв [12]. Далее признаки с последнего слоя нейронной сети можно подать в другой классический алгоритм [25]. В примере ниже представлен слой условных случайных полей (CRF Layer), в качестве признаков к которому подаются веса с выхода последнего слоя двунаправленной нейронной сети (рис. 4).

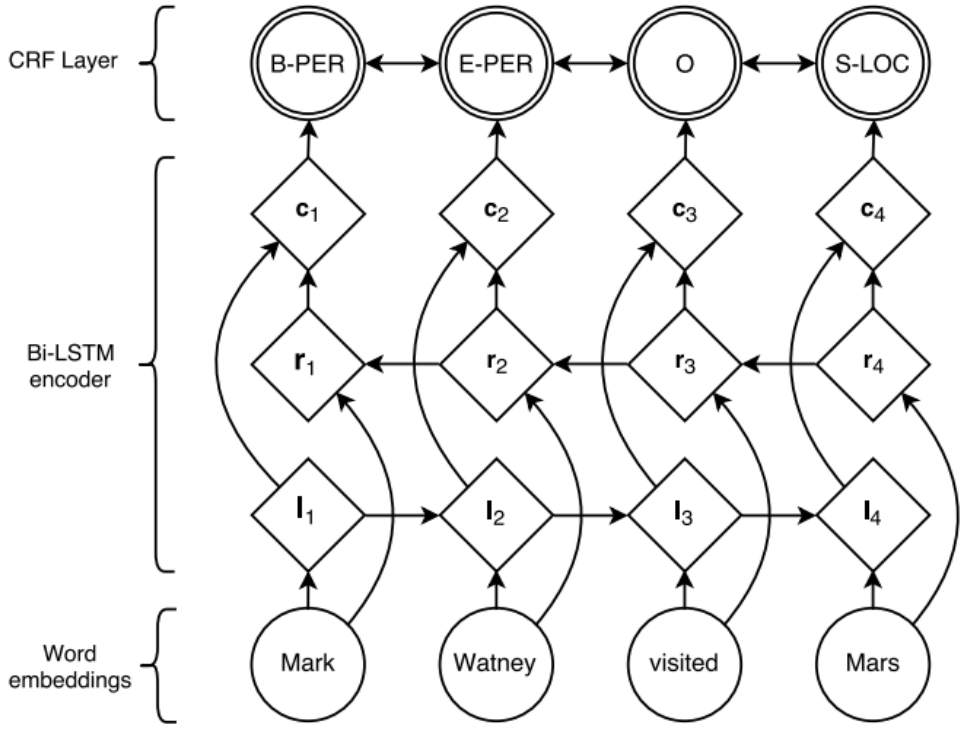

Рис. 4. Пример шаблонного решения на примере нейронной сети

Очень популярны гибридные системы, которые могут сочетать в себе все типы методов для решения задачи распознавания именованных сущностей (рис. 5). Стандартным компонентом для решения задач естественного языка являются

предобученные на неразмеченных данных векторные представления слов (word embeddings на рис. 4). В настоящее время бурно развиваются языковые модели [18], выходы с которой используются в качестве входных признаков для решения различных задач. Языковая модель – это глубокая нейронная сеть, которая учится предсказывать следующее слово по данной ей последовательности предыдущих слов, обучение происходит на большом корпусе неразмеченных данных. Такая модель содержит в себе глубокое понимание грамматики и других элементов естественного языка.

Также популярны векторные представления слов, полученные из дистрибутивных семантических моделей: word2vec, fastText [15]. Такие модели кодируют в себе смысловое значение слов (семантика), но дают единственное контекстно-независимое представление.

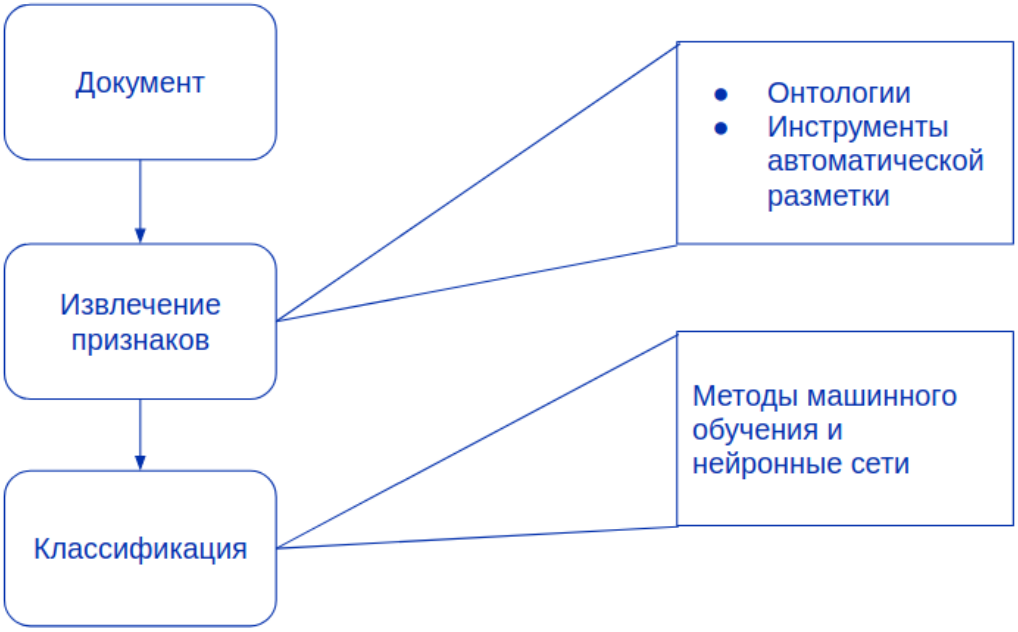

Рис. 5. Схема гибридной системы

Выводы по главе 1

В области обработки естественного языка решается много задач и используются различные методы. В настоящее время очень популярны подходы

с использованием глубоких нейронных сетей, так как они позволяют достичь наилучших результатов. Важностью частью является выбор векторного представления слов. Выделяются векторные представления, полученные из дистрибутивной семантической модели или языковой модели.

ГЛАВА 2. ОПИСАНИЕ АЛГОРИТМОВ И АРХИТЕКТУР

В данной главе будет дано описание всех алгоритмов и архитектур нейронных сетей, которые использовались в практических экспериментах. Будут описаны различные векторные представления слов, которые несут в себе необходимые знания естественного языка. Также будут описаны подходы обучения с частичным привлечением учителя и техника тонкой настройки весов, которая необходима для достижения высоких результатов в решении специфических доменных задач.

2.1. LONG SHORT-TERM MEMORY РЕКУРРЕНТНАЯ НЕЙРОННАЯ СЕТЬ

Рекуррентные нейронные сети используются для решения большого числа задач, включая задачи обработки естественного языка, из-за их способности учитывать предыдущую информацию из последовательности для расчёта текущего выхода. Однако было обнаружено [32], что, несмотря на теоретическую возможность изучать долгосрочную зависимость, на практике модели рекуррентных нейронных сетей не выполняют ожидаемого и страдают от проблем с градиентным спуском. По этой причине была разработана специальная архитектура нейронной сети под названием Long Short-Term Memory (LSTM) для решения проблемы затухания градиента [33]. LSTM представляет собой блок памяти, который содержит следующие компоненты: input gate, output gate, forget gate and memory cell. Формулы компонентов:

$$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + b_i),$$
$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f),$$
$$c_n = g(W_{cx}x_t + W_{ch}h_{t-1} + b_c),$$
$$c_t = f_t \circ c_{t-1} + i_t \circ c_n,$$
$$h_t = o_t \circ g(c_t),$$
$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + b_o),$$

где $\sigma -$ сигмоидная функция, $g -$ гиперболический тангенс, $W -$ матрица весов, $b -$ байесовские вектора и $i, f, o, c -$ компоненты блока памяти.

2.2. Двунаправленная LSTM

Правильное распознавание именованного объекта в предложении зависит от контекста слова. Как предыдущие, так и следующие слова имеют значение для прогнозирования определённой именованной сущности. Двунаправленные рекуррентные нейронные сети (bi-LSTM) [34] были разработаны для кодирования каждого элемента в последовательности с учетом левого и правого контекстов, что делает их одним из лучших вариантов для решения задачи распознавания именованных сущностей. Двунаправленный расчет модели состоит из двух этапов: прямой слой вычисляет представление левого контекста, и обратный слой вычисляет представление правого контекста. Выходы этих шагов затем объединяются для получения полного представления элемента входной последовательности. Было показано, что bi-LSTM полезны во многих задачах естественного языка, таких как машинный перевод, ответ-вопросные системы и особенно в распознавании именованных сущностей.

2.3. Модель случайных условных полей

Условное случайное поле (CRF) - вероятностная модель для структурированного предсказания, которая успешно применяется в различных областях, таких как компьютерное зрение, биоинформатика, обработка естественного языка. CRF может использоваться независимо для решения задачи распознавания именованных сущностей [25].

CRF модель обучается, чтобы предсказывать вектор $y = \{y_0, y_1, \dots, y_T\}$ тегов данного предложения $x = \{x_0, x_1, \dots, x_T\}$. Для этого вычисляется условная вероятность:

$$p(y|x) = \frac{e^{Score(x,y)}}{\sum_{y'} e^{Score(x,y')}},$$

где $Score$ вычисляется следующей формулой:

$$Score(x, y) = \sum_{i=0}^{T} A_{y_i, y_{i+1}} + \sum_{i=1}^{T} P_i, y_i ,$$

где $A_{y_i, y_{(i+1)}}$ обозначает вероятность, которая представляет собой оценку перехода от тега i к тегу j, $P_{i,j}$ - вероятность перехода, которая представляет оценку j-го тега слова i . На этапе обучения логарифмическая вероятность правильной последовательности меток $log(p\ (y \mid x))$ максимизируется.

2.4. Объединённая модель bi-LSTM и CRF

Русский язык – это морфологически и грамматически богатый язык. Таким образом, можно предположить, что сочетание CRF-модели с выходными весами нейронной сети bi-LSTM должно повысить точность выделенных именованных сущностей [12]. Пример архитектуры модели показан на рисунке 4 в пункте 1.2.2. Для каждого входного предложения сеть bi-LSTM выдает последовательность оценок, которые представляют вероятность каждого тега для каждого слова. Для повышения точности предсказаний CRF слой [25] обучается с ограничениями, зависящими от порядка тегов.

2.5. Векторное представление слов

Как уже отмечалось выше, стандартным компонентом нейронной сети для решения задач обработки естественного языка являются предобученные векторные представления слов. В настоящее время очень популярны дистрибутивные семантические модели, которые дают единственное контекстно-независимое представление слова [1, 2]. Такие модели не решают проблему полисемии. В работе исследованы глубокие контекстно-зависимые представления слов, взятые из обученной двунаправленной языковой модели и

их комбинация с дистрибутивными представлениями. Также для повышения точности предсказаний проводится тонкая настройка весов (fine tuning) двунаправленной языковой модели на текстах, которые используются для обучения модели распознавания именованных сущностей. При тонкой настройке весов в двунаправленной языковой модели нам всё также нужны неразмеченные данные. Однако, тонкую настройку весов следует проводить аккуратно, чтобы нейронная модель не потеряла знания, выделенные при первоначальном обучении.

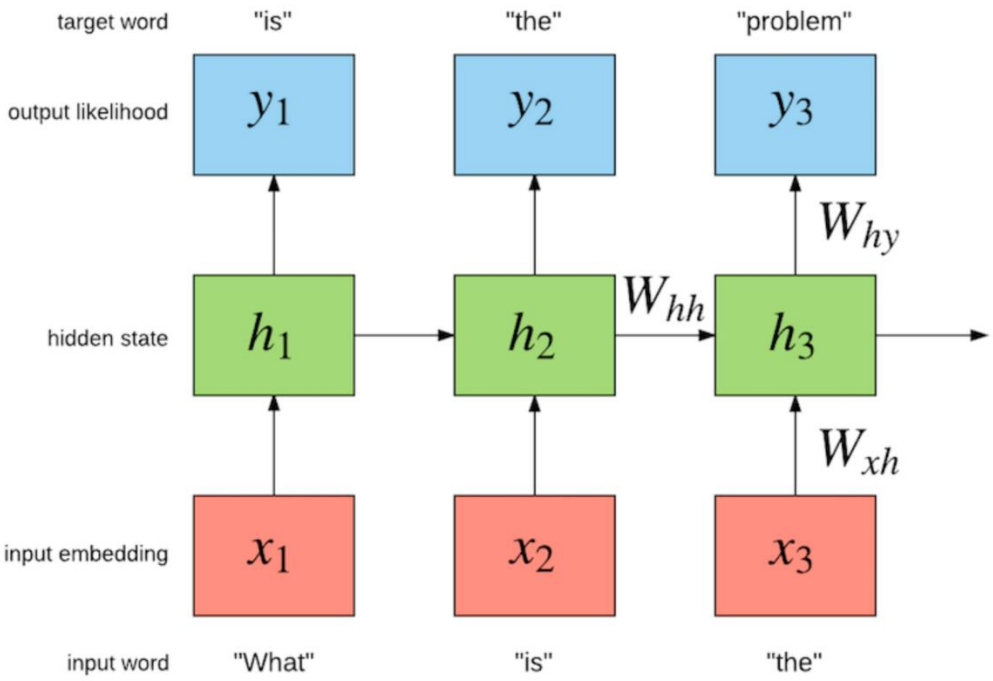

Рис. 5. Языковая модель, представленная простой рекуррентной сетью.

2.5.1. Двунаправленная языковая модель

Языковая модель по данной последовательности слов предсказывает какое слово будет следующим. На рисунке 5 показана архитектура простой рекуррентной сети, которая является языковой моделью.

Дана последовательность из N токенов, $(t_1, t_2, \ldots, t_N)$, прямая языковая модель вычисляет вероятность последовательности моделированием вероятности токена t_k по предыдущим токенам $(t_1, \ldots, t_{k-1})$:

$$p(t_1, t_2, \ldots, t_N) = \prod_{k=1}^{N} p(t_k | \, t_1, t_2, \ldots, t_{k-1})$$

Недавние современные языковые модели [18] вычисляют контекстно-независимое представление токена x_k^{LM} (через буквенное представление или через представление свёрточной нейронной сети) потом передают его в прямую рекуррентную сеть LSTM с L слоями. В каждой позиции k, каждый LSTM слой даёт на выходе контекстно-зависимое представление $\overrightarrow{h_{k,j}^{LM}}$ где $j = 1, \ldots, L$. Выход с верхнего LSTM слоя, $\overrightarrow{h_{k,L}^{LM}}$, используется для предсказания следующего токена t_{k+1} вместе со слоем Softmax (обобщение логистической функции для многомерного случая). Обратная языковая модель похожа на прямую языковую модель, в том лишь отличии, что она обрабатывает последовательность токенов в обратном направлении, предсказывая предыдущий токен по данным следующим токенам. Это может быть реализовано аналогично прямой языковой модели. Для каждого обратного LSTM слоя j модель выдает контекстно-зависимое представление $\overleftarrow{h_{k,j}^{LM}}$ токена t_k по данным токенам $(t_{k+1}, \ldots, t_N)$.

Двунаправленная языковая модель сочетает в себе как прямую, так и обратную языковую модель. Эта формулировка совместно максимизирует логарифмическую вероятность прямого и обратного направлений [14]:

$$\sum_{k=1}^{N} (\log p(t_k | t_1, \ldots, t_{k-1}; \, \Theta_x, \overrightarrow{\Theta_{LSTM}}, \, \Theta_S) + \log \, p(t_k | \, t_{k+1}, \ldots, t_N; \, \Theta_x, \overleftarrow{\Theta_{LSTM}}, \Theta_s))$$

Для прямой и обратной языковой моделей параметры (Θ_x) and Softmax слой (Θ_s) объединяются, пока в то же время остальные параметры разделены. В

целом эта формулировка аналогична подходу [9], за исключением того, что некоторые весы разделяются между направлениями вместо использования полностью независимых параметров.

2.5.2. Тонкая настройка двунаправленной языковой модели

Для тонкой настройки мы используем подход, подобный в этой работе [19]. Мы используем дискриминационную тонкую настройку и постепенное размораживание, техники для сохранения прежних знаний и избегания катастрофического забывания при тонкой настройке.

Дискриминативная тонкая настройка. Поскольку различные слои кодируют различные типы информации [20], их следует настраивать в различной степени. Для этого мы используем дискриминативную тонкую настройку. Вместо того, чтобы использовать ту же скорость обучения для всех слоев модели, дискриминативная тонкая настройка позволяет нам настраивать каждый уровень с разными скоростями обучения. Регулярное обновление стохастическим градиентным спуском (SGD) параметров модели θ на временном шаге t выглядит следующим образом [21]:

$$\theta_t = \theta_{t-1} - \eta \cdot \nabla_\theta J(\theta)$$

где η - скорость обучения, а $\nabla_\theta J(\theta)$ - градиент в отношении целевой функции модели. Для дискриминативной тонкой настройки мы разбиваем параметры θ на $\{\theta^1, \ldots, \theta^L\}$, где θ^l содержит параметры модели на l-м слое, а L - количество слоев модели. Аналогично получаем $\{\eta_1, \ldots, \eta_L\}$, где η_l - скорость обучения l-го слоя.

SGD обновление с дискриминативной настройкой заключается в следующем:

$$\theta_t^l = \theta_{t-1}^l - \eta^l \cdot \nabla_{\theta l} J(\theta)$$

Практика показывает, что при выборе скорости обучения на последнем слое η_L скорость обучения на других слоях рассчитывается следующим образом $\eta^{l-1} = \frac{\eta^l}{2.6}$.

Постепенное размораживание. Вместо того, чтобы тонко настраивать все слои одновременно, что рискованно из-за катастрофического забытья, мы используем постепенное размораживание модели, начиная с последнего слоя, поскольку он содержит наименьшее общее знание [20]. Сначала мы размораживаем последний слой и тонко настраиваем все незамерзшие слои за одну эпоху. Затем мы размораживаем следующий нижний замороженный слой и повторяем до тех пор, пока не закончим все слои. Это похоже на «chain-thaw» [22], за исключением того, что мы добавляем слой каждый раз к множеству «оттаиваемых» слоев, а не обучаем каждый слой отдельности.

2.5.3. Глубокое контекстно-зависимое представление слов

Глубокие контекстно-зависимые представления слов получаются из комбинации выходов промежуточных слоёв двунаправленной языковой модели (ELMo embeddings). Сочетание внутренних состояний даёт очень богатые репрезентации слов. Последние слои двунаправленной языковой модели кодируют контекстно-зависимые смысловые значения слова, в то время как более ранние слои моделируют синтаксис (например, они могут быть используется для разметки по частям речи). Одновременное объединение всех этих сигналов очень полезно. Для каждой конкретной задачи позволяет выбирать наиболее лучшую комбинацию.

Для каждого токена t_k, двунаправленная языковая модель с L слоями вычисляет множество из $2L + 1$ векторных представлений.

$$R_k = \{x_k^{LM}, \overrightarrow{h_{k,j}^{LM}}, \overleftarrow{h_{k,j}^{LM}} \mid j = 1, \ldots, L\} = \{h_{k,j}^{LM} \mid j = 0, \ldots, L\},$$

где $h_{k,0}^{LM}$ слой представления токена и $h_{k,j}^{LM} = [\overrightarrow{h_{k,j}^{LM}}; \overleftarrow{h_{k,j}^{LM}}]$, для каждого bi-LSTM слоя. Для включения в следующую модель, ELMo трансформирует R в один вектор, $ELMo_k = E(R_k; \Theta_e)$. В простейшем случае, ELMo просто выбирает верхний слой, $E(R_k) = h_{k,L}^{LM}$, как в TagLM [9] и CoVe [8]. В более общем смысле, векторное представление слов получается из всех слоёв двунаправленной языковой модели:

$$ELMo_k = E(R_k; \Theta) = \gamma \sum_{j=0}^{L} s_j \, h_{k,j}^{LM}. \quad (1)$$

В (1) s являются softmax-нормированными весами, а скалярный параметр γ позволяет модели для конкретной задачи масштабировать весь вектор ELMo. γ имеет практическое значение для содействия процессу оптимизации. Учитывая, что активации каждого слоя двунаправленной языковой модели имеют разное распределение, в некоторых случаях также помогает применить нормализацию на каждом слою [23].

2.6. Архитектура двунаправленной языковой модели

В этой статье мы использовали AllenNLP tensorflow-реализацию двунаправленной языковой модели [17]. Эта архитектура двунаправленной языковой модели похожа на архитектуру в [18], но модифицирована для поддержки совместного обучения обоих направлений и добавления остаточного соединения (residual connection) между слоями LSTM [38]. Архитектуру сети можно увидеть на рисунке 7.

Чтобы сбалансировать качество модели с размером модели и вычислительными требованиями для последующих задач при сохранении чисто входного представления на основе символов, были изменены все входные и

внутренние размерности в лучшей модели CNN-BIG-LSTM в [18]. Конечная модель использует $L = 2$ слоя bi-LSTM размерностью 2048 и 512 размерными

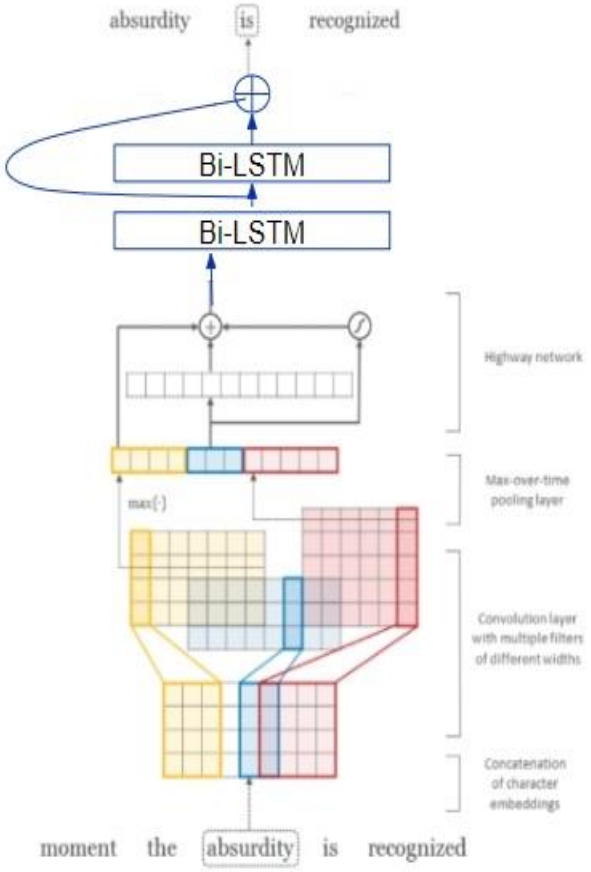

Рис. 7. Архитектура нейронной сети, которая используется в двунаправленной языковой модели

проекциями. Также используется residual connection [38] между первым и вторым слоем. Контекстно-независимое представление использует 2048 свёрточных фильтров, за которыми следует два highway слоя [24] и линейная проекция до 512 размерного представления.

2.7. Архитектура рекуррентной нейронной сети с многослойным перцептроном

Для первоначального решения задачи распознавания именованных сущностей была спроектирована архитектура рекуррентной нейронной сети с многослойным персептроном. Было взято 25 различных морфологических и

выделенных вручную признаков (лемма, часть речи, префикс, постфикс, граммема, на какой позиции стоит в предложении, является ли первым словом в предложении, является ли последним и другие). В соответствии с построенным на каждом предложении синтаксическим деревом мы дополнили представление слова, представлением предка и предка предка. Также дополнили признаками окружающих слов с окном в пять слов. В итоге получилось 825 размерное представление, которое идёт на вход рекуррентной нейронной сети, состоящей из трёх слоёв LSTM и трёх полносвязных слоёв (многослойный персептрон). Также применяется вариационный дропаут [39] для предотвращения переобучения и достижения лучших результатов. Архитектуру нейронной сети можно увидеть на рисунке 6.

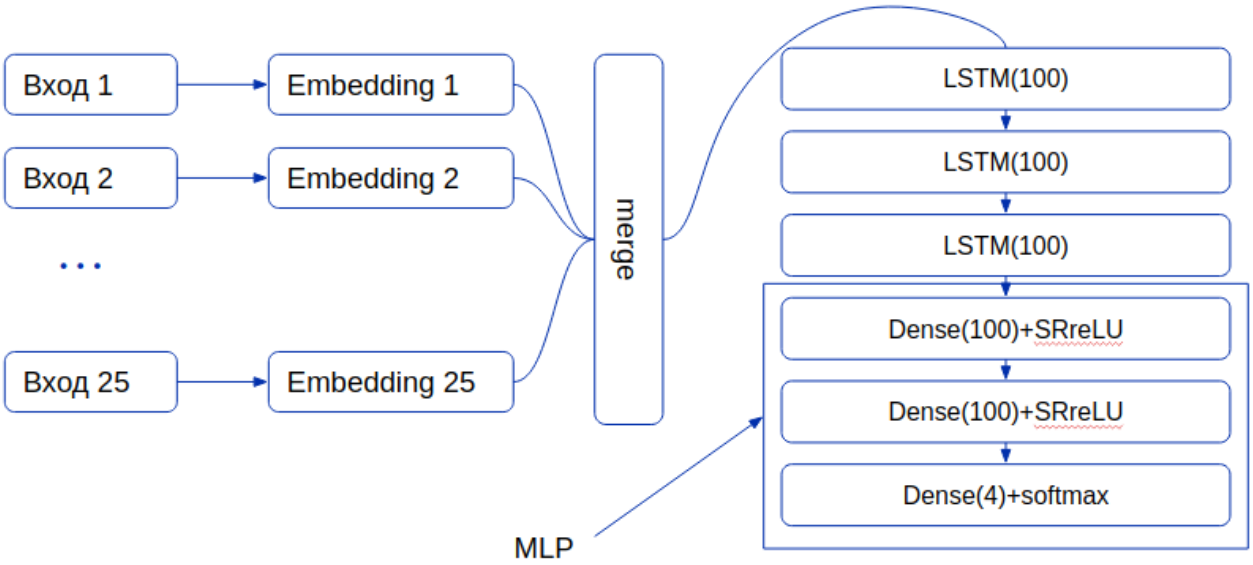

Рис. 6. Рекуррентная нейронная сеть + МСП.

2.8. Архитектура двунаправленной рекуррентной нейронной сети

Следуя последним новейшим системам [9, 12], базовая модель для решения задачи распознавания именованных сущностей использует

предварительно обученные представления слов, два слоя bi-LSTM, полносвязный слой, выходы с которого идут на вход алгоритму условных случайных полей (CRF) [25], аналогично [3]. Два слоя bi-LSTM, первый из которых содержит 40 нейронов, а второй - 20 нейронов, полносвязный слой содержит число нейронов равное числу тегов (число именованных сущностей + 1). Во время тестирования выполняется декодирование с использованием алгоритма Витерби. Также применяется вариационный дропаут [39] для предотвращения переобучения и достижения лучших результатов. Архитектуру нейронной сети можно увидеть на рисунке №7.

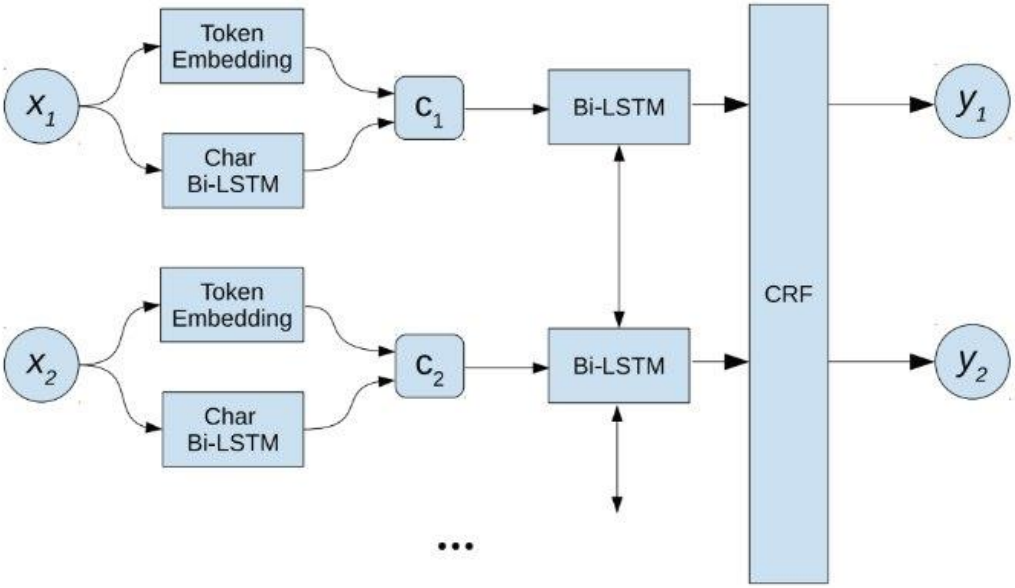

Рис. 7. Архитектура двунаправленной рекуррентной нейронной сети с CRF слоем.

Выводы по главе 2

Из-за простоты и эффективности предобученные векторные представления слов стали широко распространены в системах обработки естественного языка. Было показано, что такие представления содержат полезную семантическую и синтаксическую информацию [1,2]. Однако, как отмечалось выше, такие представления допускают только одно контекстно-

независимое представление слова. Обучение различных представлений для разных смысловых значений одного и того же слова – трудная задача.

Ранее были предложены методы, которые решают некоторые проблемы традиционного подхода, дополняя его информацией части слова [4, 5], или изучая различные векторные представления для разных смыслов одного слова [6]. Другая недавняя работа также была посвящена изучению контекстно-зависимых представлений. Context2vec [7] использует двунаправленную Long Short Term Memory (LSTM) для кодирования контекста вокруг конкретного слова. Другие подходы для получения контекстно-зависимых представлений слов используют выходы с последнего рекуррентного слоя языковой модели [9].

В настоящее время текущие современные модели тегирования последовательности обычно включают в себя двунаправленную рекуррентную нейронную сеть (RNN), которая кодирует последовательности слов в контекстно-зависимое представление, прежде чем делать предсказания, специфичные для слов [11, 12, 13]. Проблема с этими моделями заключается в том, что они обучаются на небольшом количестве размеченных данных и не полностью учитывают контекст каждого слова. В работе [11] были изучены методы совместного обучения двунаправленной рекуррентной нейронной сети с дополнительными размеченными данными из других задач.

В данной работе представляется альтернативный подход с частичным обучением без учителя, который не требует дополнительных размеченных данных. Используется глубокое контекстно-зависимое представление слова, полученное от языковой модели, обученной на большом корпусе неразмеченных данных. Также использование тонкой настройки языковой модели позволяет добиться лучшей точности предсказаний, а настройка происходит также на неразмеченных данных. Глубокие контекстно-зависимые представления легко объединяются с дистрибутивными представлениями и используются в решении конкретных задач естественного языка.

Главная цель работы - показать, что контекстно-зависимое представление, полученное из внутренних состояний языковой модели, в

комбинации с дистрибутивным представлением полезно в задачи распознавания именованных в русском языке и позволяет достичь самых лучших результатов.

ГЛАВА 3. ПРАКТИЧЕСКОЕ ИССЛЕДОВАНИЕ

Все практические исследования проводились на языке Python 3.5. Для проведения экспериментов и реализации спроектированных архитектур нейронных сетей использовались библиотеки машинного обучения Keras и Tensorflow. Обучение нейронных сетей происходило на сервере университета ИТМО, который имеет 4 видеокарты GeForce GTX 1080 12Gb. Время обучения нейронных сетей зависит от размера данных и размера самой нейронной сети. В работе например, двунаправленная рекуррентная нейронная сеть с CRF слоем обучалась на 30 эпохах с использованием одного GPU в течение часа, а вот двунаправленная языковая модель с использованием трёх GPU обучается около 10 дней.

3.1. НАБОР ДАННЫХ

В работе использован FactRuEval корпус данных. Корпус состоит из новостных и аналитических текстов на русском языке, посвященных социальным и политическим вопросам. Тексты были собраны из следующих источников:

- private Correspondent (http://www.chaskor.ru/);
- wikinews (https://ru.wikinews.org).

Корпус был разделен на две части: демонстрационный корпус из 122 текстов и тестовый корпус из 133 текстов. Демонстрационный корпус имеет около 30 тысяч токенов и 1700 предложений. Тестовый корпус имеет около 60 тысяч токенов и 3100 предложений. Модели нейронных сетей для решения задачи распознавания именованных сущностей обучались на демонстрационном корпусе и тестировались на тестовом корпусе. Необходимо было классифицировать каждый токен на следующие классы: person, location, organization, остальные (O) (классические именованные сущности, глава 1).

3.2. Методы и метрики оценки качества

Оценка качества систем проводится на корпусах, размеченных вручную. На серии конференций CoNLL (Conference on Computational Natural Language Learning) был предложен простой способ оценки: именованная сущность выделена системой правильно, если ее класс и границы, обозначенные системой, совпадают с классом и границами, размеченными в корпусе; иначе сущность выделена неправильно. Точность (precision, P), полнота (recall, R) и $F1$-мера в данном случае определяются следующим образом:

$$Precision = \frac{\text{число верно выделенных сущностей}}{\text{число всех выделенных сущностей}} \quad (1)$$

$$Recall = \frac{\text{число верно выделенных сущностей}}{\text{число сущностей в корпусе}} \quad (2)$$

$$F1 = \frac{2PR}{P + R} \quad (3)$$

Системы по распознаванию именованных сущностей можно оценивать немного по-другому. Так, на конференциях MUC (Message Understanding Conference) качество работы систем измерялось по двум составляющим: способности правильно распознать тег именованной сущности или выделить правильно её границы. В зависимости от требований в конкретной задаче значимость составляющих можно определять по-разному, смотря что важнее: распознать наличие сущности, или правильно выделить именно границы.

В работе для оценки качества моделей используются скрипты, написанные специально для проведения соревнования. Метрики такие же как (1), (2), (3) – micro F1 score.

3.3. Обучение нейронной сети на морфологических и синтаксических признаках

Первоначальным предположением было то, что для решения задачи распознавания именованных сущностей на вход рекуррентной нейронной сети необходимо подать такое представление слов, которое содержит в себе морфологические и синтаксические признаки. Векторное представление слова строилось на основе признаков рядом стоящих слов и слов, являющихся предками в синтаксическом дереве разбора. В итоге получается 33 слова:

- '' - текущее слово
- '2dom' — родитель родителя текущего слова
- '1dom' — родитель текущего слова
- '-5' — 5ое слово слева от текущего («-» - слева, «+» - справа)
- '-5_2dom' — родитель родителя 5-го слова слева от текущего
- '-5_1dom' — родитель «-5го»

И 25 различныхпризнаков для каждого слова: prefix4 prefix3 prefix2 prefix1 postfix4 postfix3 postfix2 postfix1 posStart pos link len lemma isupper istitle islower isdigit isalpha isalnum isLastWord isFirstWord grm forma dom ID.

Как отмечалось в пункте 2.7 всего на вход получается 825 признаков. Архитектура сети описана также в пункте 2.7. Полный набор параметров для этой модели состоит из параметров каждого слоя нейронной сети (весовые матрицы, смещения, матрицы векторных представлений слов). Все эти параметры настраиваются во время тренировки по алгоритму обратного распространения ошибки со стохастическим градиентным спуском (lr = 0.1, decay = 1e-7, momentum = 0, clipvalue = 3).

После обучения 30 эпох модель RNN + MLP (пункт 2.7) достигает precision – 72.90, recall – 76.60 и F1 – 74.71. Входные признаки кодируют

недостаточно хорошо знания естественного языка, и модель обучается на небольшом количестве рамеченных данных. Для лучших результатов необходимо глубокое понимание грамматики, семантики и других элементов естественного языка, чего нет в полной мере во взятых признаках.

3.4. Обучение двунаправленной рекуррентной нейронной сети с CRF слоем и дистрибутивным векторным представлением слов

Дальнейшим предположением было добавить на вход представление слов, взятое из дистрибутивной семантической модели. Дистрибутивные семантические модели кодируют смысловое значение слов. Эти признаки мы объединяли с некоторыми морфологическими признаками и подавали на вход двунаправленной рекуррентной нейронной сети с CRF слоем (пункт 2.8).

В качестве дистрибутивных векторных представления были взяты представления из дистрибутивной семантической модели fastText, предобученной на текстах Common Crawl [30] и Wikipedia [31]. Эти модели были обучены с использованием CBOW алгоритма, размерностью 300, с буквенными n-граммами длины 5, окном размером 5.

Полный набор параметров для этой модели состоит из параметров слоев bi-LSTM (весовых матриц, смещений, матрицы векторных представлений слов) и матрицы перехода слоя CRF. Все эти параметры настраиваются во время тренировки по алгоритму обратного распространения со стохастическим градиентным спуском (lr = 0.1, decay = 1e-7, moemntum = 0, clipvalue = 3). Variational Dropout применяется для предотвращения переобучения и улучшения производительности модели. После обучения на 30 эпохах модель bi-LSTM + CRF достигает precision – 77.23, recall – 85.19 и F1 – 81.02.

3.5. Обучение двунаправленной рекуррентной нейронной сети с CRF слоем и ELMo представлением слов.

В этой главе описаны эксперименты с глубоким контекстно-зависимым представлением слов (ELMo).

3.5.1. Обучение двунаправленной языковой модели

Мы обучаем модель biLM на большом корпусе данных. Архитектура сети описана в пункте 2.6, в работе использована AllienNLP tensorflow реализация [17]. Данные взяты из conll2017. После подготовки данных получается около 200 миллионов токенов и 15 миллионов предложений, словарь составляет 60 тысяч наиболее часто встречающихся слов. После обучения в течение 5 эпох перплексия составляет около 45. После предварительного обучения biLM может вычислять векторные представления слов для любой задачи. Для последующего обучения модели двунаправленной рекуррентной сети для решения задачи распознавания именованных сущностей были сформированы ElMo представления слов, полученные из biLM. Полное множество параметров при обучении языковой модели:

```
options = {
        'batch_size': 64,
        'n_gpus': 3,
        'bidirectional': True,
        'char_cnn': {'activation': 'relu',
         'embedding': {'dim': 16},
         'filters': [[1, 32],
          [2, 32],
          [3, 64],
          [4, 128],
          [5, 256],
```

```
            [6, 512],
            [7, 1024]],
           'max_characters_per_token': 50,
           'n_characters': 261,
           'n_highway': 2},

          'dropout': 0.1,
          'lstm': {
           'cell_clip': 3,
           'dim': 4096,
           'n_layers': 2,
           'proj_clip': 3,
           'projection_dim': 512,
           'use_skip_connections': True},
          'all_clip_norm_val': 10.0,
          'n_epochs': 10,
          'n_train_tokens': n_train_tokens,
          'batch_size': batch_size,
          'n_tokens_vocab': vocab.size,
          'unroll_steps': 20,
          'n_negative_samples_batch': 8192,
}
```

3.5.2. Тонкая настройка двунаправленной языковой модели

В некоторых случаях тонкая настройка biLM на данных, относящиеся к задаче, приводит к значительным падениям перплексии и увеличению производительности последующих моделей. Для точной настройки модели мы используем неразмеченные предложения демонстрационного корпуса FactRuEval, которые также были доступны во время соревнования. BiLM обучается на 3 эпохах и оценивается по предложениям тестового корпуса.

Перплексия на тестовov корпусе после тонкой настройки упала с 90 до 40 (чем меньше, тем лучше). После тонкой настройки biLM были сформированы ELMo представления слов, которые подаются на вход другой модели.

3.5.3. Результаты с ELMo представлениями

Дистрибутивные векторные представления слова (fastText) объединяются с глубокими контекстно-зависимыми представлениями (ELMo) до тонкой настройки biLM и подаются на вход bi-LSTM сети с CRF слоем. После обучения на 30 эпохах модель Bi-LSTM + CRF достигает precision – 82.32, recall – 84.04 и F1 – 83.17.

Заключительным шагом дистрибутивные векторные представления слова (fastText) объединяются с глубокими контекстно-зависимыми представлениями (ELMo) после тонкой настройки biLM и подаются на вход bi-LSTM сети с CRF слоем. После обучения на 30 эпохах модель Bi-LSTM + CRF достигает precision - 83.19, recall - 85.41 и F1 - 84.29. Результаты показаны в таблице 1.

3.6. Сравнение результатов

Традиционные подходы к распознаванию именованных сущностей в русском языке в значительной степени основывались на ручных правилах и внешних ресурсах. Так в работе [40] применялись регулярные выражения и словари для решения задачи. Следующим шагом было применение методов статистического обучения, таких как условные случайные поля (CRF) и поддерживающие векторные машины (SVM) для классификации сущностей. CRF над лингвистическими признаками, рассматриваемыми как базовый уровень в исследовании [41]. В работе [42] предложили двухэтапный алгоритм условных случайных полей: на первом этапе на вход для CRF подавались вручную выделенные лингвистические признаки. Затем на втором те же

лингвистические признаки были объединены с глобальной статистикой, рассчитанной на первом этапе и подавались в CRF. В работе [27] применяли классификацию SVM к дистрибутивным векторынм представлениям слов и фраз. Эти представления были получены путем обучения без учителя дистрибутивных семантических моделей на различных новостных корпусах данных. Одновременное использование словарных признаков и распределенных векторных представлений слов было представлено в [28]. Словарные признаки были извлечены из Wikidata, а векторные представления были предварительно предобучены на данных из Википедии. Затем эти объединённые признаки использовались для классификации с помощью SVM. В настоящий момент методы обучения с использованием глубоких нейронных сетей являются наиболее перспективным вариантом для NER. В работе [43] предложена глубокая нейронная сеть LSTM для решения задачи распознавания именованных сущностей на русском языке. Отличительной особенностью этой работы является связывание задачи моделирования языка с классификацией именованных сущностей, что очень похоже на подход, использующийся в данной выпускной работе.

В исследовании [26] применяется современная модель нейронной сети для английского NER к открытым размеченным для NER на русском языке корпусам данным. Модель состоит из трех основных компонентов: двунаправленные рекуррентные сети (bi-LSTM), CRF и распределенное векторное представление слов. Эксперименты показывают, что только Bi-LSTM несколько хуже, чем модель с CRF [41]. Добавление CRF в качестве следующего этапа обработки поверх слоёв Bi-LSTM значительно улучшает производительность модели и позволяет превзойти модель, представленную в [41]. Различие модели bi-LSTM + CRF модели [26], от модели представленной в [41], является обученные векторные представления слов и фраз. Обучение сети Bi-LSTM еще и на уровнях символов даёт лучшие результаты, чем ручные признаки в [41].

Как уже отмечалось дистрибутивные векторные представления стали стандартным инструментом в области обработки естественного языка. Такие представления способны захватывать семантические особенности слов и значительно улучшать результаты для различных задач, что продемонстрировано в работе [26].

Авторы работы [26] полагают, что результаты LSTM highway network могут быть улучшены путем лучшей инициализации весов и более глубокой архитектурой. Кроме того, укладка слоев highway LSTM может улучшить результаты, позволяющие сети динамически корректировать сложность обработки.

Результаты всех описанных подходов на корпусе данных FactRuEval показаны в таблице 1.

	Precision	Recall	F1-score
Rubaylo [29]	77.70	78.50	78.13
Wiki-based aproach [28]	88.19	64.75	74.67
basic+dictionary + w2v features on SVM [28]	82.57	74.08	78.10
SVM+w2v [27]	-	-	82.88
Bi-LSTM + CRF + Lenta [26]	83.80	80.84	82.10
NeuroNER + Highway char [26]	80.59	80.72	80.66
Stanford Core NLP	71.01	64.19	67.13
RNN+MLP	72.90	76.60	74.71
Bi-LSTM-CRF	77.23	85.19	81.02
Bi-LSTM-CRF + ELMo	82.32	84.04	83.17
Bi-LSTM-CRF+ ELMo + fine tuning	83.19	85.41	84.29

Таб. 1. Сравнительная таблица результатов. Последние четыре строки – эксперименты, описанные выше, остальные результаты других работ.

Выводы по главе 3

В этой работе мы показали, что контекстно-зависимое представление, полезно в задаче распознавания именованных сущностей в русском языке. Когда мы тщательно настроили нашу двунаправленную языковую модель и включили такие представления, мы достигаем самых современных результатов.

Также в работе показано повышение точности модели с постепенной её адаптацией, добавлением различных знаний морфологии, синтаксиса и семантики. В дальнейшем планируется более тщательная работа над обучением языковой модели, тонкой настройки и получением контекстно-зависимым векторных представлений.

ЗАКЛЮЧЕНИЕ

В данной работе была достигнуты наилучшие результаты в решении задачи распознавания именованных сущностей в русском языке. Сама задача в мире уже решается давно и на различных языках. В английском языке сейчас самые высокие результаты. Около двадцати лет назад начинали решать задачу с помощью рукописных правил, постепенно усложняя алгоритмы, применяя статистические модели. Сейчас во всех языках с этой задачей лучше всего справляются глубокие двунаправленные рекуррентные нейронные сети. Их успех продиктован многими факторами. Относительно задачи распознавания именованных сущностей можно выделить то, что для объекта действительно важен контекст с обеих сторон. Для нейронной сети всё по-прежнему важно для обучения иметь большой набор данных. В доменных задачах часто это бывает невозможным, поэтому приходится находить другие способы извлечения знаний из естественного языка. В данной работе был представлен метод обучения с частичным привлечением учителя, когда другая модель обучается на большом корпусе неразмеченных данных и знания от неё можно передать модели, которая будет решать конкретную задачу, но обучаться на небольшом количестве размеченных данных. В нашем случае это была двунаправленная языковая модель, которая выучивает грамматику языка. Ранее было показано, что на разных ей слоях кодируются различные знания. В начале работы приводилась пирамида задач в естественном языке. Также и в нейронных сетях: на нижнем слое морфология, а на верхнем – семантика, где-то межу находится синтаксис. Часто в решении доменных задач бывают проблемы с присутствием специфических слов и выражений, которых не было при обучении моделей без учителя. Для борьбы с такими словами языковую модель обучают, к примеру, на частях слова или на буквах. Также применяют тонкую настройку нейронной сети, что извлечь знания из доменной области. Однако, тонкую настройку необходимо проводить аккуратно, чтобы в последствии доменная модель не переобучалась и сохраняла обобщающую способность: на неизвестных при обучении данных тоже показывала высокие результаты.

В русском языке есть ещё направления, в которых можно развивать решение задачи распознавания именованных сущностей. Не до конца исследованы языковые модели. Нет чёткого понимания между качеством обучения языковой модели и точностью в решении доменных задач, то есть не понятно, когда же языковая модель обучилась лучше и ей применение даст лучшие результаты на наследуемой модели.

В дальнейшем предполагается продолжить изучение языковых моделей. Обучение их в русском языке, эксперименты с различными архитектурами, получение глубоких контекстно-зависимых представлений. Ведь, такие представления полезны во многих задачах обработки естественного языка. С каждым годом решается ряд трудностей в обработке естественного языка и человечество на шаг становится ближе к созданию сильного искусственного интеллекта.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Mikolov T. et al. Distributed representations of words and phrases and their compositionality //Advances in neural information processing systems. – 2013. – С. 3111-3119.
2. Pennington J., Socher R., Manning C. Glove: Global vectors for word representation //Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). – 2014. – С. 1532-1543.
3. Collobert R. et al. Natural language processing (almost) from scratch //Journal of Machine Learning Research. – 2011. – Т. 12. – №. Aug. – С. 2493-2537.
4. Wieting J. et al. Charagram: Embedding words and sentences via character n-grams //arXiv preprint arXiv:1607.02789. – 2016.
5. Bojanowski P. et al. Enriching word vectors with subword information //arXiv preprint arXiv:1607.04606. – 2016.
6. Neelakantan A. et al. Efficient non-parametric estimation of multiple embeddings per word in vector space //arXiv preprint arXiv:1504.06654. – 2015.
7. Melamud O., Goldberger J., Dagan I. context2vec: Learning generic context embedding with bidirectional lstm //Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning. – 2016. – С. 51-61.
8. McCann B. et al. Learned in translation: Contextualized word vectors //Advances in Neural Information Processing Systems. – 2017. – С. 6297-6308.
9. Peters M. E. et al. Semi-supervised sequence tagging with bidirectional language models //arXiv preprint arXiv:1705.00108. – 2017.
10. Belinkov Y. et al. What do Neural Machine Translation Models Learn about Morphology? //arXiv preprint arXiv:1704.03471. – 2017.
11. Yang Z., Salakhutdinov R., Cohen W. W. Transfer learning for sequence tagging with hierarchical recurrent networks //arXiv preprint arXiv:1703.06345. – 2017.
12. Lample G. et al. Neural architectures for named entity recognition //arXiv preprint arXiv:1603.01360. – 2016.
13. Hashimoto K. et al. A joint many-task model: Growing a neural network for multiple NLP tasks //arXiv preprint arXiv:1611.01587. – 2016.
14. Peters M. E. et al. Deep contextualized word representations //arXiv preprint arXiv:1802.05365. – 2018.
15. https://github.com/facebookresearch/fastText
16. https://github.com/dialogue-evaluation/factRuEval-2016
17. https://github.com/allenai/bilm-tf
18. Jozefowicz R. et al. Exploring the limits of language modeling //arXiv preprint arXiv:1602.02410. – 2016.
19. Howard J., Ruder S. Fine-tuned Language Models for Text Classification //arXiv preprint arXiv:1801.06146. – 2018.
20. Yosinski J. et al. How transferable are features in deep neural networks? //Advances in neural information processing systems. – 2014. – С. 3320-3328.
21. Ruder S. An overview of gradient descent optimization algorithms //arXiv preprint arXiv:1609.04747. – 2016.

22. Felbo B. et al. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm //arXiv preprint arXiv:1708.00524. – 2017.
23. Ba J. L., Kiros J. R., Hinton G. E. Layer normalization //arXiv preprint arXiv:1607.06450. – 2016.
24. Srivastava R. K., Greff K., Schmidhuber J. Training very deep networks //Advances in neural information processing systems. – 2015. – С. 2377-2385.
25. Lafferty J., McCallum A., Pereira F. C. N. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. – 2001.
26. Anh L. T., Arkhipov M. Y., Burtsev M. S. Application of a Hybrid Bi-LSTM-CRF model to the task of Russian Named Entity Recognition //arXiv preprint arXiv:1709.09686. – 2017.
27. Ivanitskiy R., Shipilo A., Kovriguina L. Russian Named Entities Recognition and Classification Using Distributed Word and Phrase Representations //SIMBig. – 2016. – С. 150-156.
28. Сысоев А. А., Андрианов И. А. Распознавание именованных сущностей: подход на основе вики-РесуРсов. 2016.
29. Rubaylo A. V., Kosenko M. Y.: Software utilities for natural language information retrievial //Almanac of modern science and education, Volume 12 (114), 87 – 92. (2016).
30. http://commoncrawl.org/
31. https://www.wikipedia.org/
32. Bengio Y., Simard P., Frasconi P. Learning long-term dependencies with gradient descent is difficult //IEEE transactions on neural networks. – 1994. – Т. 5. – №. 2. – С. 157-166.
33. Hochreiter S., Schmidhuber J. Long short-term memory //Neural computation. – 1997. – Т. 9. – №. 8. – С. 1735-1780.
34. Schuster M., Paliwal K. K. Bidirectional recurrent neural networks //IEEE Transactions on Signal Processing. – 1997. – Т. 45. – №. 11. – С. 2673-2681.
35. Krogh A. et al. Predicting transmembrane protein topology with a hidden markov model: application to complete genomes1 //Journal of molecular biology. – 2001. – Т. 305. – №. 3. – С. 567-580.
36. Safavian S. R., Landgrebe D. A survey of decision tree classifier methodology //IEEE transactions on systems, man, and cybernetics. – 1991. – Т. 21. – №. 3. – С. 660-674.
37. Chapelle O., Scholkopf B., Zien A. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews] //IEEE Transactions on Neural Networks. – 2009. – Т. 20. – №. 3. – С. 542-542.
38. Szegedy C. et al. Inception-v4, inception-resnet and the impact of residual connections on learning //AAAI. – 2017. – Т. 4. – С. 12.
39. Kingma D. P., Salimans T., Welling M. Variational dropout and the local reparameterization trick //Advances in Neural Information Processing Systems. – 2015. – С. 2575-2583.

40. Trofimov I. V. Person name recognition in news articles based on the persons-1000/1111-F collections //16th All-Russian Scientific Conference Digital Libraries: Advanced Methods and Technologies, Digital Collection, RCDL. – 2014. – С. 217-221.
41. Gareev R. et al. Introducing baselines for Russian named entity recognition //International Conference on Intelligent Text Processing and Computational Linguistics. – Springer, Berlin, Heidelberg, 2013. – С. 329-342.
42. Mozharova V., Loukachevitch N. Two-stage approach in Russian named entity recognition //Intelligence, Social Media and Web (ISMW FRUCT), 2016 International FRUCT Conference on. – IEEE, 2016. – С. 1-6.
43. Malykh V., Ozerin A. Reproducing Russian NER Baseline Quality without Additional Data //CDUD@ CLA. – 2016. – С. 54-59.